

PATENT ABSTRACTS OF JAPAN

(11)Publication number : **07-066834**
(43)Date of publication of application : **10.03.1995**

(51)Int.Cl.

H04L 12/56

(21)Application number : **06-154988**
(22)Date of filing : **06.07.1994**

(71)Applicant : **INTERNATL BUSINESS MACH CORP <IBM>**
(72)Inventor : **GALAND CLAUDE**
SCOTTON PAOLO

(30)Priority

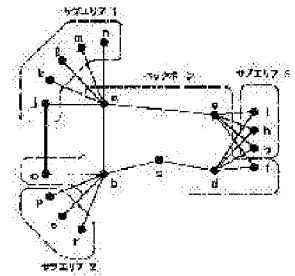
Priority number : **93 93480105** Priority date : **30.07.1993** Priority country : **EP**

(54) COMMUNICATION NETWORK AND PATH SELECTION METHOD FOR THE SAME

(57)Abstract:

PURPOSE: To speed up the routing path search by determining a designated route for each request of connection between an access node and a destination node out of the set of usable links automatically selected in advance for each destination node positioned inside a network.

CONSTITUTION: Inside a packet exchange communication network, a routing controller A prepares a connective list and a tree from the number of connection links of respective nodes (a)-(r) while using a network topology. Besides, respective nodes (a)-(f) and (g)-(r) are classified into the backbone and geographical subareas 1-3 and stored in a data base together with the connective list and tree by the device A. Next, corresponding to each request of connection between the access node and the destination node, the designated path is calculated from the list and tree stored by the device A, determined and connected to a packet exchange device. Thus, the routing path search can be speeded up without degrading the optimization criterion of the routing algorithm and without generating any additional control messages.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-66834

(43) 公開日 平成7年(1995)3月10日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
H 0 4 L 12/56		9077-5K	H 0 4 L 11/ 20	1 0 2 D

審査請求 有 請求項の数12 O L (全 23 頁)

(21) 出願番号 特願平6-154988

(22) 出願日 平成6年(1994)7月6日

(31) 優先権主張番号 9 3 4 8 0 1 0 5 . 1

(32) 優先日 1993年7月30日

(33) 優先権主張国 フランス (F R)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州アーモンク (番地なし)

(72) 発明者 フロード・ガラン

フランス06800 カーニュ・シュル・メール アヴェニュー・ド・テュイリエール56

(74) 代理人 弁理士 合田 潔 (外2名)

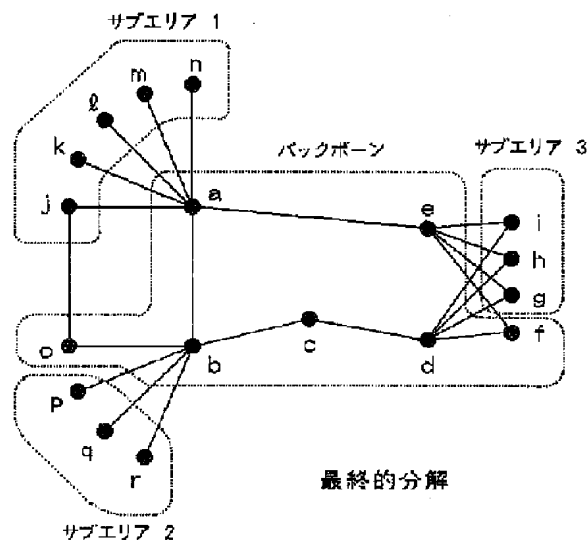
最終頁に続く

(54) 【発明の名称】 通信ネットワークおよび該ネットワークでの経路選択方法

(57) 【要約】

【目的】 本発明の目的は、経路指定アルゴリズムの最適化基準を低下させず、またネットワーク上に追加の制御メッセージを生成せずに、パケット交換ネットワークのバックボーン・ノード及びサブエリア・ノードへの自動的分解を実行して、指定経路探索を高速化することである。

【構成】 高スループット回線によって相互接続された1組のノードを使って、高度なメッシュ構成のバックボーンを構築し、次いでローカル・ノードを、それ自体がバックボーンに接続された地理的サブエリアにグループ化する。経路指定アルゴリズムはこの特定のネットワーク・トポロジーを利用して、経路計算の複雑さを劇的に削減する。所与の接続について、限られた数のノードだけが使用可能であると定義され、その経路計算の際にアルゴリズムによって考慮される。



【特許請求の範囲】

【請求項1】 伝送リンクと相互接続された複数のノードを含むパケット交換通信ネットワークにおいて、該ネットワークのネットワーク・アクセス・ノードが、データ・パケットを受信し送信する手段と、ネットワーク構成を記憶する手段とを含むとともに、ネットワーク内に位置する各宛先ノード用の1組の使用可能リンクを自動的に事前選択する手段と、前記使用可能リンクの事前選択情報を局所的に記憶する手段と、前記アクセス・ノードと宛先ノードの間の接続要求ごとに、前記事前選択された使用可能リンクから指定経路を決定する手段とを含む、通信ネットワーク。

【請求項2】 前記リンク事前選択手段が、前記ネットワークを1組のバックボーン・ノードと複数のサブエリア・ノードとに分解する手段を含む、請求項1に記載のネットワーク。

【請求項3】 前記分解手段が、前記記憶手段内の前記ネットワーク構成にアクセスする手段と、

各ネットワーク・ノードに接続されたリンクの数である接続性に応じて、すべてのネットワーク・ノードを分類する手段と、

すべてのネットワーク・ノードを含む非循環グラフであって、各ノードが一度だけ現れ、その起点（ルート）が最高の接続性をもつノードのうちから選ばれる、接続性ツリーを構築する手段と、

前記接続性ツリーの終端ノードではないノードをバックボーン・ノードにし、前記接続性ツリーの分離した終端を構成するノード（終端だけを「子」として有する該終端の「親」）を該バックボーンに含ませ、同じサブエリアに属する同じ「親」を有する、バックボーン・ノードではないノードをサブエリア・ノードにして、前記ノードをバックボーン・ノードとサブエリア・ノードに分類する手段と、

2つのサブエリア間のすべてのリンクを検出し、該リンクのそれぞれを接続する2つのノードのうちで接続性が最高のノードをバックボーンに組み入れる手段と、

バックボーンからサブエリアへのリンクを分析し、接続性ツリーにおける前記サブエリア・ノードの「親」がバックボーン・ノードに接続されるという条件と、前記サブエリア・ノードの「親」がバックボーン・ノードの「親」に接続されるという条件とを両方とも満たさないサブエリア・ノードを決定し、同じサブエリアにおける前記条件を満たさないすべてのノードのうちの1つを事前に定義された基準に従ってバックボーンに組み入れる手段と、

空のサブエリア、もしくは事前定義されたノード数より少ないノードを含むサブエリアまたはその両方を除去する手段とを含む、請求項2に記載のネットワーク。

【請求項4】 前記リンク事前選択手段が、

2つのバックボーン・ノードを接続するリンクであるバックボーン・リンクを決定する手段と、同じサブエリア内の2つのノードまたはサブエリア・ノードとバックボーン・ノードを接続するリンクである、サブエリア・リンクを決定する手段とを含む、請求項2または3に記載のネットワーク。

【請求項5】 前記リンク事前選択手段が、

(i) 宛先ノードがアクセス・ノードと同じサブエリア内に位置するときはアクセス・ノード・サブエリアに属するリンク、(ii) 宛先ノードもしくはアクセス・ノードの一方がバックボーン内に位置するときは該当するノードのサブエリアに属するリンクおよびバックボーン・リンク、(iii) アクセス・ノードと宛先ノードが共にバックボーン内に位置するときはバックボーン・リンク、(iv) アクセス・ノードと宛先ノードが異なるサブエリア内に位置するときはバックボーン・リンクとアクセス・ノード・サブエリアに属するリンクと宛先ノードに属するリンクを、経路決定に使用可能なものとして選択する手段と、

他のすべてのリンクを、経路決定に使用可能でないものとして選択する手段とを含む、請求項4に記載のネットワーク。

【請求項6】 前記指定経路決定手段が、修正ベルマン・フォード・アルゴリズムを計算する手段を含む、請求項1、2、3、4または5に記載のネットワーク。

【請求項7】 アクセス・ノード（300）において、伝送リンク（209）と相互接続された複数のノード（201ないし208）とを含むパケット交換通信ネットワーク（200）内で指定経路を選択する方法であって、ネットワーク構成を記憶する段階を含み、

ネットワーク内に位置する各宛先ノードのために1組の使用可能リンクを自動的に事前選択する段階と、

前記使用可能リンクの事前選択情報を局所的に記憶する段階と、

前記アクセス・ノードと宛先ノードの間の接続要求ごとに、前記事前選択された使用可能リンクから指定経路を決定する段階とを含む方法。

【請求項8】 前記事前選択段階が、前記ネットワークを1組のバックボーン・ノードと複数のサブエリア・ノードとに分解する段階を含む、請求項7に記載の方法。

【請求項9】 前記分解段階が、

前記記憶手段内の前記ネットワーク構成にアクセスする段階と、

各ネットワーク・ノードに接続されたリンクの数である接続性に応じて、すべてのネットワーク・ノードを分類する段階と、

すべてのネットワーク・ノードを含む非循環グラフであって、各ノードが一度だけ現れ、その起点（ルート）が最高の接続性をもつノードのうちから選ばれる、接続性

ツリーを構築する段階と、

前記接続性ツリーの終端ノードではないノードをバックボーン・ノードにし、前記接続性ツリーの分離した終端を構成するノード（終端だけを「子」として有する該終端の「親」）を該バックボーンに含ませ、同じサブエリアに属する同じ「親」を有する、バックボーン・ノードではないノードをサブエリア・ノードにして、前記ノードをバックボーン・ノードとサブエリア・ノードに分類する段階と、

2つのサブエリア間のすべてのリンクを検出し、該リンクのそれぞれを接続する2つのノードのうちで接続性が最高のノードをバックボーンに組み入れる段階と、バックボーンからサブエリアへのリンクを分析し、接続性ツリーにおける前記サブエリア・ノードの「親」がバックボーン・ノードに接続されるという条件と、前記サブエリア・ノードの「親」がバックボーン・ノードの「親」に接続されるという条件を両方とも満たさない、サブエリア・ノードを決定し、同じサブエリアにおける前記条件を満たさない、すべてのノードのうちの1つを事前に定義された基準に従ってバックボーンに組み入れる段階と、

空のサブエリアもしくは事前定義されたノード数より少ないノードを含むサブエリアまたはその両方を除去する段階と、

を含む、請求項8に記載の方法。

【請求項10】前記リンク事前選択段階が、2つのバックボーン・ノードを接続するリンクであるバックボーン・リンクを決定する段階と、同じサブエリア内の2つのノードまたはサブエリア・ノードとバックボーン・ノードを接続するリンクであるサブエリア・リンクを決定する段階とを含む、請求項8または9に記載の方法。

【請求項11】前記リンク事前選択段階が、

(i) 宛先ノードがアクセス・ノードと同じサブエリア内に位置するときはアクセス・ノード・サブエリアに属するリンク、(ii) 宛先ノードもしくはアクセス・ノードの一方がバックボーン内に位置するときは該当するノードのサブエリアに属するリンクおよびバックボーン・リンク、(iii) アクセス・ノードと宛先ノードが共にバックボーン内に位置するときはバックボーン・リンク、(iv) アクセス・ノードと宛先ノードが異なるサブエリア内に位置するときはバックボーン・リンクとアクセス・ノード・サブエリアに属するリンクと宛先ノードに属するリンクを、経路決定に使用可能なものとして選択する段階と、

他のすべてのリンクを、経路決定に使用可能でないものとして選択する段階とを含む、請求項10に記載の方法。

【請求項12】前記経路決定段階が、修正ベルマン・フォード・アルゴリズムを計算する段階を含む、請求項

7、8、9、10または11に記載の方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、大規模パケット交換ネットワークにおける高速伝送システムに関し、より詳細には、ネットワーク・トポロジをバックボーンと複数のサブエリアとに自動的に分解し使用可能リンクを事前選択する手段によって、ソース・ノードと宛先ノードの間の経路選択を高速化するための装置および方法に関する。

【0002】

【従来の技術】データを「パケット」と呼ばれるデータの集合体として伝送するパケット交換ネットワーク手段によって、複数のデータ処理要素を相互接続することがますます有用になってきた。そのようなネットワークは、データ処理装置を支援するエンド・ノードに接続された複数の相互接続された交換ノードを含んでいる。そのようなパケット・ネットワークは、地理的に広く分布した非常に大きなものになり得る。そのような状況では、互いに通信しようとする2つのエンド・ノード間の効率的な経路選択が最も重要になる。H. ヌスバウマー(Nussbaumer)のTeleinformatique2 (pp. 92~117)"Presses Polytechniques Romandes 1987"にさまざまな方法が要約されている。

【0003】経路選択：ネットワーク・レベルで解決しなければならない一般的な問題は、ソース・ノードと宛先ノードの間の経路を見つけることである。ネットワークがデータグラムを使用するときは、経路選択は各パケットごとに個別に行わなければならない。仮想回線では、経路決定は回路確立時にだけ1度行われる。どちらの場合にも、経路指定アルゴリズムの選択は、しばしば相反する多くの必要条件を満たさなければならないので、容易ではない。このアルゴリズムは、ノードの実現が複雑にならないように実施できるよう簡単でなければならない、ネットワーク上にどんな混乱があっても、正確なパケットの進行を保証しなければならない。トラフィックの変動やネットワーク・トポロジのいかにかわらず、このアルゴリズムは満足できる結果を提供できなければならない。また、さまざまなエンド・ユーザが公平になるようにし、ネットワークにアクセスする権利を平等に分配しなければならない。最後に、経路指定アルゴリズムは、できれば利用タイプによって変わり得る基準に従って、ネットワークを最適に利用できるようにしなければならない。多くの場合、ネットワークは、パケット伝送時間が最短になり、かつ最大数のパケットが転送できるように実施される。その主な目的は、それぞれ伝送時間の削減とスループットの向上である。他の場合には、通信コストを削減し、あるいは破局的な回線、ノード故障、トラフィック・ピークの場合にも正確に動作（過度に性能が低下することなく）できる信頼性の高い

ネットワークを開発することがその目的である。さまざまな制約があるため、多数の異なるタイプの経路指定がある。そのあるものは、トラフィックの変動およびネットワーク・トポロジを統合するそれぞれの能力に応じて決定の方針 (deterministic policy: 固定方針) または適応の方針 (adaptive policy) に対応する。特定のノードから経路が決定される場合は、経路指定アルゴリズムを集中化することができる。他のものは全ノードに分散される。これは、信頼性の点では好ましいが、アルゴリズムがより複雑になり、経路の最適化を実施することがより難しくなる。いくつかのアルゴリズムは分類が難しいが、あふれ経路指定 (flooding routing) またはランダム経路指定 (random routing) と呼ばれる特別な技法を使用している。

【0004】 1. あふれ経路指定

あふれ経路指定は、各ノードが、入力リンクを除くそのノードのすべての出力リンク上にパケットを再送信するという非常に簡単な原理に基づくものである。K個の他のノードに接続されたノードが、受け取ったばかりのパケットのK-1個のコピーを再送信する。この方法により、ある条件でパケットの少なくとも1つのコピーを宛先ノードへ配布することが可能になり、したがってソース・ノードから宛先ノードに至る少なくとも1つの経路が存在しなければならない。この経路指定は、たとえばネットワーク内のいくつかの構成要素の破局的障害の後でネットワーク・トポロジが変化するときでも、確実に10 行われる。あふれ経路指定は、ネットワーク利用において非常に強い頑丈さを与える。さらに、ソース・ノードと宛先ノードの間の考えられるリンクがすべて網羅的に試され、ネットワークが過負荷でない場合、この方法は、少なくとも1つのパケットのコピーが最小の遅延でより短い経路を通して受取り側に到達するようにする。この技法は、メッセージを経路指定するための経路指定テーブルも、ネットワークにおける宛先ノードの地理的位置の知識も必要ないので、簡単に実施することができる。受取り側は、パケットが自己宛にアドレス指定されることだけを知っていればよい。

【0005】 これらの頑丈さ、簡易さおよび敏速さの品質に対する代償として、ネットワーク資源の利用度が低下し、また混雑が生じやすくなる。通常、パケット・ヘッダ内に、パケットが通過することを許されるノードの最大数を示すフィールドを置くことが好都合である。このフィールドは、パケットがノードを通過するたびに減分され、空のフィールドを有するパケットが破棄される。

【0006】 2. ランダム経路指定または確率的 (stochastic) 経路指定

この技法は、あふれ経路指定とは異なり、ノード自体のレベルで経路決定を下すためのネットワーク構造およびトラフィックについての完全な知識を有することをノードに要求しない。しかしながら、余りに大きなゴースト

・トラフィックの発生を制限するために、ノードはそれ自体が受け取ったパケットのコピーをすべての出力リンク上に系統的に送ることはしない。ランダム経路指定は、任意に選択した出力リンク上に受信パケットの1つまたは複数のコピーを放出するものである。この方法は、選択的あふれとも呼ばれる。この方針は、ノード・レベルにおける非常に簡単な経路選択をもたらす、パケットの複製によるネットワークの飽和を制限する。しかし、平均経路指定遅延は決定的手法による場合よりも長くなる。パケットは、宛先ノードに向かう最短の経路ではなく遠回りの経路をとる。

【0007】 3. 決定経路指定

(a) 固定式経路指定

固定式経路指定は、ネットワーク・トポロジや通信回線上で予想される平均トラフィックなど一般のネットワーク特性にしたがって、異なるパケットがとるべき経路を定義する。経路指定規則は1回確立され、ユーザが優先する性能基準の最適化を狙いとする。ほとんどの場合、ネットワークを介する平均パケット伝送時間を最小にしなければならない。この方法は、各ノードのレベルで経路指定テーブルを作成するものである。協議により、受け取ったばかりのパケットをどの出力リンク上に転送しなければならないかを、ノードが決定できるようになる。固定式経路指定は、その原理が非常に簡単である。ノード内で行われる処理は待ち行列の管理とテーブルの読み取りだけであり、最適化アルゴリズムはネットワーク設計レベルで1回だけ始動される。固定式経路選択は、ネットワークの長期的かつ総合的な最適化を狙いとするが、ランダム経路選択と比較すると、ネットワークを介するパケット伝送速度をかなり高めることができる。

【0008】 (b) 適応的経路指定

固定式経路指定とは反対に、適応的経路指定の目的はいかなるときも最適化基準を満たすことである。たとえば回線上のトラフィックの瞬間的な状態に応じて、上述のテーブルが絶えず更新される。

【0009】 ・集中経路指定

ネットワーク特性が絶えず変化するときは、トラフィックの変動およびトポロジに応じて経路指定テーブルを周期的に更新する責任をあるノードに割り当てることによって、経路指定を適応化することができる。この方法は、原理的には単純であり、集中経路指定と呼ばれる。その主な欠点は、大きな補助トラフィックを発生すること、およびネットワークの良好な機能をただ1つのノードに従属させてしまうことである。さまざまなノードが、それらが知覚したネットワークの状態 (動作可能な隣接ノード、トラフィック...) を経路指定センターに送らなければならない。経路指定センターは各ノードにそれらの経路指定テーブルをディスペッチしなければならない。補助トラフィックは経路指定センターの周囲に集中し、このためネットワークのこの部分が飽和することに

なる。さらに、テーブルがすべてのノードによって同時に受け取られることはないので、集中経路指定はテーブルがリフレッシュされるときにいくつかの問題を引き起こす可能性がある。1つの解決策は、テーブルの確立を各ノード・レベルに分散させることである。

【0010】・ローカル経路指定

ローカル経路指定方法も分散式経路指定方法（後述）も共に、局所的に収集された情報に従って、各ノードによってそれ自体の経路指定テーブルを確立することに基づくものである。ローカル経路指定技法では、各ノードが情報を隣接ノードと交換することなくそのテーブルを作成する。最も簡単な形では、この方法は、受け取ったばかりのパケットをより短い出力待ち行列内に置き、それをできるだけ迅速に伝送することからなる。ローカル・アルゴリズムは原理的に、ネットワーク内でパケットを非常に素早く循環させる傾向がある。ただし、経路はさまざまな方法で選択されるので、平均経路長は最小値とは大きく異なる。

【0011】・分散式経路指定

分散式経路指定は、経路指定テーブルと遅延テーブルを更新するために、隣接ノードがトラフィックおよびネットワーク状態に関するメッセージを交換する局所的経路指定方法である。

【0012】・階層式経路指定

非確率的経路指定方法では、各ノードは、そのネットワーク内のノードと同数の行と、出力リンクの数と等しい数の列とを有するテーブルを必要とする。さらに、経路指定が適応式のとき、ノードは経路指定テーブルを更新するためにメッセージを周期的に交換しなければならない。テーブル・サイズおよび制御メッセージの大きさが、ノード数とともに急激に増大し、ネットワークが10個を超えるノードを含むと許容できなくなる。この問題を解決するために、ネットワークを、図1に示すように、サブネットワークの階層に分割する。具体的には、必要な経路指定テーブルの長さが最小になるように、最適クラスタ構造（003）を決定する。前記テーブルはサブネットワーク（003）内のノードだけを考慮しており、別のサブネットワーク内のノードにアドレス指定されたパケットは、必然的にこの遠隔サブネットワークのアクセス・ノード（005）を通過する。パケットはいくつかの必須ノード（005）を通過し、この単純化の代償は、経路指定テーブルが小さくなることにより、経路指定情報が正確でなくなり、その結果メッセージ・トラフィックの経路長が長くなることである。階層的経路指定手続きは、大規模パケット交換ネットワーク（001）で特に効果的である。経路指定テーブルが小さくなると、ノード（002）内で必要な記憶域と処理が少なくなり、かつ回線（004）における通信オーバーヘッドが少なくなる。

【0013】階層レベルの最適化の問題は、Computer N

etworks, Vol.1, pp.155~174 (1977年1月)に所載の、L. クラインロック(Kleinrock)およびF. カムーン(Kamoun)の論文"Hierarchical Routing for Large Networks"で検討されている。

【0014】経路指定テーブル長を削減するための主な概念は、任意のノードにおいて、近接したノードに関する完全な経路指定情報と、遠隔に位置するノードに関するより少ない情報を維持することである。これはテーブルに、近接ノードの宛先ごとに1つのエントリを提供し、遠隔ノードの1組の宛先ごとに1つのエントリを提供することによって実現される。

【0015】経路指定テーブル長の削減は、ネットワークの階層区分によって達成される。基本的に、1組のノード（002）のmレベルの階層的クラスタ化（図1）は、ノードを第1レベルのクラスタ（003）にグループ化し、次にこれを第2レベルのクラスタにグループ化し、以下同様にグループ化を続けることからなる。この動作は下から上へと続けられる。第mレベルのクラスタは、最高レベルのクラスタ（第3レベルのクラスタ）であり、したがってネットワーク（001）のすべてのノードを含む。

【0016】あいにく、テーブル長のこれらの利得はネットワーク内のメッセージ経路長の増加を伴う。その結果、より長い経路長によって引き起こされる過度の内部トラフィックのために、ネットワーク性能（遅延、スループット）が低下する。しかし、極めて大きいネットワークでは、ネットワーク経路長の著しい増加なしに、大きなテーブルの縮小が達成される。

【0017】高性能ネットワーク：データ伝送は、アプリケーションに特に焦点を合わせ、また顧客のトラフィック・プロファイルに基本シフトを組み込むことにより進展してきている。ワークステーション、ローカル・エリア・ネットワーク（LAN）相互接続、ワークステーションとスーパー・コンピュータの間の分散処理、新しいアプリケーション、および階層ネットワークと対等ネットワーク、広域ネットワーク（WAN）とローカル・エリア・ネットワーク、音声とデータなどしばしば相反する異なる構造の統合の成長により、データ・プロファイルは、帯域幅が大きくなり、パースト的、非確率論的になり、またより高い接続性を必要とする。上記のことから、チャネル接続されたホスト、事務およびエンジニアリング・ワークステーション、端末装置並びに小さいものでは中間ファイル・サーバの間でLAN通信、音声、ビデオおよびトラフィックを搬送できる高速ネットワークの分散計算アプリケーションを支援することが強く必要とされていることは明らかである。高速マルチプロトコル・ネットワークのこの展望が、データ、音声、およびビデオ情報がデジタル・コード化され、小さなパケットに分断され、共通の1組のノードとリンクを介して伝送される、高速パケット交換ネットワーク・アー

キテクチャの出現の原動力となっている。

【0018】1. スループット

これらの新しいアーキテクチャの重要な要件は、実時間配布の制約を満たし、音声およびビデオの移送に必要なノードの高スループットを達成するために、端末間の遅延時間を減少させることである。リンク速度の増大に比例して、通信ノードの処理速度が増大しないと、高速ネットワークの基本課題は、各ノード内のパケット処理時間を最小にすることである。たとえば、2つのエンド・ユーザ間で音声パケットを伝送するために、一般的な1000ミリ秒の遅延を満たすには次のことが必要である。

【0019】・エンド点におけるパケット化およびプレイ・アウト機能に、合計36ミリ秒必要である。

【0020】・たとえばアメリカ合衆国を横断するのに必要な固定伝播遅延時間は、およそ20ミリ秒である。

【0021】・ネットワーク中をパケットが移動する際の全ノード内処理時間として44ミリ秒残る。5ノードのネットワークでは、待ち行列時間を含めた全処理時間として各ノードは約8ミリ秒有する。10ノードのネットワークでは、各ノードは約4ミリ秒有する。

【0022】図2では、同じ制約に対する別の見方が示されている。1MIPS（毎秒100万回の命令）の有効処理速度を有するノードを取り上げてみると、処理されるパケットごとにネットワークが833000個の命令を実行しなければならない場合でも、9.6kbpsの回線を1000バイトのパケットで満たすことができる。64kbpsの回線では、ノードはパケットごとに125000回の命令を収容できる。しかしながら、OC24リンクを満たすためには、この1MIPSのノードは、パケットごとに7個の命令しか実行できない。後者の場合は、10～30MIPSの有効処理速度でも、パケットごとに70～200回の命令しか収容できない。

【0023】処理時間を最小にし、高速／低エラー率の技法を完全に利用するために、新しい高帯域ネットワーク・アーキテクチャによって提供される転送および制御機能の大部分は、端末間方式で実行される。フロー制御および特に経路選択は、中間ノードの知識と機能の両方を減少させるネットワークのアクセス点によって管理される。

【0024】2. 経路指定モード

ネットワーク内の経路指定は、次の2つの側面をもつ。

1. 所与の接続のための経路を決定する。

2. 交換ノード内でパケットを実際に交換する。

【0025】ネットワークを介する経路の決定方法は数多くある。極めて高いスループットを得るために重要なことは、着信パケットをどこに経路指定するかを、交換エレメントが極めて短時間内に判断できなければならないことである。"High Speed Networking Technology, An Introductory Survey" pp. 88～96, GG24-3816-00 ITSC R

eleigh March 1992に述べられているように、最もよく使用される経路指定モードは、ソース経路指定（ソース・ルーティング）およびラベル・スワッピングである。

【0026】・ソース経路指定

ソース経路指定は、無接続ネットワーク用の分散経路指定の特定の実施態様である。ソース・ノード（またはアクセス・ノード）が、パケットがネットワークを介して取るべき経路を計算する責任を負う。経路指定フィールドが送られたすべてのパケットに付加され、中間ノードがそのフィールドを使ってパケットを宛先に導く。この技法では、送信ノードがネットワーク・トポロジを知らなければならない。最適経路を見つけることができない。しかし経路が決定された後は、中間ノードは経路指定を行うためにどのテーブルやパラメータも参照する必要がない。経路指定の次の段階はまさにパケット・ヘッダで行われる。この方法の欠点は、パケット・ヘッダ内の経路指定フィールドが、ある記憶域を占有し、オーバーヘッドとなることである。しかしこれはまったく些細なことであり、高速に経路指定を決定できるという利点が、帯域オーバーヘッドのわずかな増加という欠点よりも勝っている。すべての経路指定情報が各パケット内に含まれるので、端末間通信を提供するために経路に沿ったセットアップを行う必要はない。したがって、データグラム・サービスから利益を受けるデータ・アプリケーションは、この技法によって有効に支援される。ただしデータグラム・トラフィックは、各データグラムを1つの接続と見ることができ、経路の計算を必要とするということも特徴とする。各パケットは、自己完結型ユニットとしてネットワークを介して経路指定され、他のすべてのパケットと無関係である。ソース・ノードに関する重要な点は、各データグラムについて、より短時間で最適の経路を決定することである。

【0027】・ラベル・スワッピング

ラベル・スワッピングは、接続指向ネットワーク用の分散経路指定の特定の実施態様である。このネットワークは、一般的に、何らかの形の「チャネル」を使用するリンク上で、多数の接続（セッション）を多重化する。リンク上を送信される各パケットは、このパケットが属する論理接続を識別する任意の番号を含むヘッダを有する。たとえば、各パケットは、そのパケットが送られる伝送リンクを識別するために中間ノードによって使用されるラベルを、その経路指定フィールド内に含むことができる。パケットがノードに到達すると、そのヘッダからラベルが抜き出され、リンク・アドレスに加えて新しいラベルを提供する経路指定テーブルを索引するために使用される。新しいラベルは古いラベルに上書きされ、パケットはそのリンク・アドレスを使用して転送される。経路指定テーブル内の情報は、接続セットアップ時に更新される。経路上の各ノードは、各接続方向用のラベルを選択し、それを隣接ノードに送る。

【0028】呼出しセットアップおよび資源予約処理は、次のような段階を含む。

【0029】・接続要求は、ユーザにより起点ネットワーク・アドレスと宛先ネットワーク・アドレスならびにデータ・フロー特性を含むパラメータ・セットを使って指定される。

【0030】・経路決定は、ソース・ノードによってその局所的経路指定トポロジ・データベースから実現される。

【0031】・経路予約は、指定されたノードに沿って、特別なメッセージに入れて送出される。このメッセージは、通常は前述のソース経路指定技法に従って経路指定され、トラフィック・タイプが必要とするサービス・レベルを提供するため、その接続テーブルをセットアップし、その資源を予約する（たとえば、経路の各リンク上の帯域幅予約）よう、ノードに指示する。

【0032】・テーブル更新は、ネットワーク資源の使用可能性の変化を反映する。その情報は、ネットワーク内のすべてのノードに送られる。

【0033】ラベル・スワッピング技法では、接続テーブルを動的にセットアップし維持する必要がある。これは、新しい接続が確立されまたは古い接続が終了されるとき、テーブルが更新されることを意味する（もちろん、ネットワーク・トポロジのデータベースは完全に独立して維持される）。これは、接続セットアップが比較的にまにしか行われず、あまり時間が重要ではない場合に可能になる。この制限のため、データグラム移送はまったく非効率的である。しかしながら、実時間音声接続のような非常に短いパケットしか必要としない接続は、パケットのオーバーヘッドが低いので、この技法により有効に支援される。接続がいったん確立された後は、パケットが送られるたびに宛先アドレスをヘッダ内に置く必要はない。このパケットのためにどの接続を使用するかを指定するフィールドだけが必要とされる。

【0034】3. 経路選択基準

パケット・ネットワーク内のノード間の経路を選択するための主な基準の1つは、最小ホップ・カウントと最小経路長である。ホップ・カウントは、2つのエンド・ノード間で経路を作成するために使用されるリンクの数である。経路長は、2つのエンド・ノード間の経路によってもたらされる全体的伝送遅延時間の関数である。大部分の高速ネットワークでは、このネットワークを介する最悪ケースの遅延がほぼ常に許容されるので、遅延（経路長）は大きな問題ではない。しかし、ホップ・カウントは、所与の経路を実施するために必要な資源量の直接の尺度であり、したがって経路を選択する際にかなり重要である。ネットワーク・リンクが混雑すると、必然的に大きいホップ・カウントの経路が選択されるので、選択された経路は最小ホップ・カウントの経路である必要がないことに留意されたい。しかし、過大な量のネット

ワーク資源が1つの経路にコミットされる結果、他の経路が一層混雑し、他の接続用に必然的にさらに長いホップ・カウントの経路が選択されることになるので、そのような長い代替経路を無制限に延ばすことは許されない。それによって、長期的なネットワークのスループットが悪影響を受ける可能性がある。したがって、問題は、最小ホップ・カウント、すなわちネットワーク資源量を過度に利用しない最小経路長を有する、起点ノードと宛先ノード間の経路を選択することである。

【0035】4. 重要な要件

接続指向経路指定モードと無接続経路指定モードのどちらもサポートする大規模高速パケット交換ネットワークにおける分散経路指定機構では、性能および資源消費量に関して、次のように要約されるいくつかの要件が暗黙に課される。

【0036】・ソース・ノード（またはソース・ノードのために経路計算を行うノード）は、着信パケットをどこに経路指定するかを非常に短期間に決定できなければならない（各接続要求のための最適経路を計算するための計算が十分に早くなければならない）。

【0037】・中間ノードにおける交換時間が最短でなければならない（最短処理時間）。

【0038】・選択された経路に沿ったネットワーク資源を、最小ホップ・カウントの基準に従って最適化しなければならない。

【0039】・ネットワークが過負荷にならないように、制御メッセージをできるだけ制限しなければならない。

【0040】高速ネットワークでは、制御メッセージの交換によって生成される経路指定テーブルを更新するためのコストは、これが接続セットアップ前に実行される限りクリティカルではない。回線スループットが極めて高いと、通信回線自体に著しい負担をかけずに経路指定情報の交換が許される。これらの考慮から、L. クラインロックとF. カムーンが提唱した階層的経路指定方法とは反対に、最適経路の基準を犠牲にせずに、ノード内で維持される経路指定テーブルのより良好なアクセスおよび使用が示唆される。

【0041】

【発明が解決しようとする課題】現在の所、経路指定アルゴリズムは、最適経路を選択する前にネットワーク内のソース・ノードから宛先ノードへの可能なすべての経路を計算する。ネットワーク・トポロジに関する仮定が行われず、経路計算はしばしば時間と資源を消費する。たとえば、ネットワークの特定の地理的構成のために、計算された経路が許容されないこともある。

【0042】現実には、大規模伝送ネットワークは完全にメッシュ化（mesh）されてはいない。本発明は、ネットワークが通常は階層構造に基づいて構築されるという単純な観察を基礎としている。ユーザにとって必要な冗

長性や信頼性が得られるようにするためには、高スループット回線で相互接続された1組のノードを使用して、高度にメッシュ化された「バックボーン」を構築する。そしてローカル・ノードを、それ自体がバックボーンに結合された地理的「サブエリア」にグループ化する。経路指定アルゴリズムは、この特定のネットワーク・トポロジを利用して、経路計算の複雑さを大幅に削減させることができる。所与の接続について、限られた数のノードだけが適格と定義され、最適経路探索の際にアルゴリズムによって考慮される。

【0043】本発明の目的は、経路指定アルゴリズムの最適化基準を下げずに、またネットワーク上に追加の制御メッセージを生成せずに経路選択を高速化するために、バックボーンおよびサブエリアへのネットワークの自動分解 (decomposition) を実行することである。

【0044】

【課題を解決するための手段】本発明は、要約すると、伝送リンクと相互接続された複数のノードとを備えるパケット交換通信ネットワーク内のアクセス・ノードにおいて、指定経路を選択するための方法であり、次のよう

- な段階を含む。
- ・ネットワーク構成を記憶する段階。この段階はさらに、下記の諸段階を含む。
- ・ネットワーク内に位置する各宛先ノードごとに1組の使用可能リンクを自動的に事前選択する段階。
- ・使用可能リンクの前記事前選択を局所的に記憶する段階。
- ・前記アクセス・ノードと宛先ノードの間の各接続要求ごとに、前記事前選択された使用可能リンクの中から指定経路を決定する段階。

【0045】

【実施例】図3に示すように、通信システムの典型的なモデルは、専用回線、キャリア提供サービスまたは公衆データ・ネットワークを使用して、高性能ネットワーク(200)を介して通信する、いくつかのユーザ・ネットワーク(212)からなる。各ユーザ・ネットワークは、企業サーバ(213)として使用される大型コンピュータ、LAN(ローカル・エリア・ネットワーク214)に接続されたワークステーションまたはパーソナル・コンピュータを使用するユーザ・グループ、アプリケーション・サーバ(215)、PBX(構内交換機216)またはビデオ・サーバ(217)を相互接続するリンク(211)ならびに通信プロセッサの組として記述される。これらのユーザ・ネットワークは、異なる施設に分散され、広域転送設備を介して相互接続する必要がある。データ転送を編成するために異なる方法が使用できる。あるアーキテクチャは、各ネットワーク・ノードでデータ保全性に関する検査を必要とし、そのため伝送が遅くなる。他のアーキテクチャは、基本的に高速データ転送を追求するもので、最終宛先に向って流れるパケ

ットができる限り最高速度で処理されるように、ノード内の経路指定技法と交換技法が最適化される。本発明は本質的に後者のカテゴリに属しており、より詳細には以下の節で詳しく述べる高速パケット交換ネットワーク・アーキテクチャに属する。

【0046】高速パケット交換ネットワーク：図3における全体図は、8個のノード(201-208)を備える高速パケット交換伝送システムを示し、各ノードはトランク(209)と呼ばれる高速通信回線によって相互接続されている。ユーザによる高速ネットワークへのアクセス(210)は、周辺に配置されたアクセス・ノード(202-205)を介して実現される。これらのアクセス・ノードは1つまたは複数のポートを含み、各ポートはそれぞれ、標準インターフェースを支援する外部装置をネットワークに接続し、ネットワークを横切ってユーザのデータ・フローを他の外部装置からまたは他の外部装置へ移送するために必要な変換を実行するためのアクセス点を提供する。たとえば、アクセス・ノード202は、構内交換機(PBX)、アプリケーション・サーバ、および3つのポートを通るハブにそれぞれインターフェースし、隣接する通過ノード201、208および205によってネットワークを介して通信する。

【0047】交換ノード：各ネットワーク・ノード(201-208)は、着信データ・パケットが発信トランク上で隣接通過ノードに向って選択的に経路指定される、経路指定点を含む。そのような経路指定の決定は、データ・パケットのヘッダ内に含まれる情報に従って行われる。基本的なパケット経路指定機能に加え、ネットワーク・ノードは次のような補助的サービスをも提供する。

- ・ノード内から発するパケット用の指定経路の決定。
- ・ネットワーク・ユーザおよび資源に関する情報の検索や更新などのディレクトリ・サービス。
- ・リンク使用状況情報を含む物理的ネットワーク・トポロジの一貫したビューの維持。
- ・ネットワークのアクセス点での資源の予約。

【0048】各ポートは複数のユーザ処理装置に接続され、各ユーザ装置は、別のユーザ・システムに伝送されるデジタル・データのソース、または別のユーザ・システムから受け取ったデジタル・データを消費するデータ受信側、あるいは通常はその両方を含む。ユーザ・プロトコルの解釈、パケット・ネットワーク(200)上で伝送するために適切にフォーマットされたパケットへのユーザ・データの変換、およびこれらのパケットを経路選択するためのヘッダの生成が、ポートで走行するアクセス・エージェントによって実行される。このヘッダは、制御フィールドと経路指定フィールドからなる。

【0049】・経路指定フィールドは、アドレス指定された宛先ノードにネットワーク(200)を介してパケットを経路指定するために必要な情報をすべて含む。

【0050】・制御フィールドは、とりわけ、経路指定フィールドを解釈する際に使われるプロトコルのコード化された識別を含む（ソース経路指定、ラベル・スワッピングなど）。

【0051】経路指定点：図4は、図3に示したネットワーク・ノード（201-208）中に見られるような、典型的な経路指定点（300）の全体的ブロック図を示す。経路指定点は、高速パケット・スイッチ（302）を含み、経路指定点に到達したパケットがそこに入る。そのようなパケットは以下のようにして受信される。

- ・高速伝送リンク（303）を経てトランク・アダプタ（304）を介して、他のノードから受信する。
- ・ポート（301）と呼ばれるアプリケーション・アダプタを介して、ユーザから受信する。

【0052】アダプタ（304、301）は、パケット・ヘッダ内の情報を使用して、どのパケットをスイッチ（交換機：302）によってローカル・ユーザ・ネットワーク（307）に向けてまたはノードから出る伝送リンク（303）に向けて経路指定するかを決定する。アダプタ（301と304）は、スイッチ（302）に入る前または後にパケットを待ち行列に入れるための待ち行列回路を含む。

【0053】経路推定制御装置（305）は、ユーザによって指定された所与の1組のサービス品質を満たし、通信経路を完成させるために用いられるネットワーク資源の量を最小にするために、ネットワーク（200）を介する最適経路を計算する。次に、経路指定点で生成されたパケットのヘッダを構築する。最適化基準には、中間ノードの数、接続要求の特徴、経路内のトランクの容量と使用度が含まれる。経路指定のために必要な、ノードおよびノードに接続された伝送リンクに関する情報はすべて、ネットワーク・トポロジ・データベース（306）内に含まれる。定常状態のもとでは、すべての経路指定点は同じネットワークのビューを有する。ネットワーク・トポロジ情報は、ネットワークに新しいノードが追加されて新しいリンクが活動化されたとき、リンクまたはノードがドロップされたとき、あるいはリンクの負荷が著しく変化したときに更新される。そのような情報は、その資源が接続されたネットワーク・ノードから発し、制御メッセージによって他のすべての経路制御装置との間で交換され、経路計算に必要な最新のトポロジ情報を提供する（そのようなデータベースの更新は、ネットワークのエンド・ユーザ間で交換されるデータ・パケットと非常に類似したパケット上で搬送される）。継続的な更新を通じてネットワーク・トポロジがあらゆるノードで最新の状態に維持されるので、エンド・ユーザの論理接続（セッション）を乱さずに、動的なネットワークの再構成が可能になる。

【0054】パケット経路指定点への着信伝送リンク

は、ローカル・ユーザ・ネットワークの外部装置からのリンク（210）、または隣接するネットワーク・ノードからのリンク（トランク：209）を含む。どちらの場合でも、経路指定点は同じように動作して、パケット・ヘッダ内の情報によって指示されるように、各データ・パケットを受け取り、別の経路指定点に転送する。高速パケット交換ネットワークは、単一パケットの伝送継続中以外はどの伝送機構やノード機構をも通信経路専用とせず、任意の2つのエンド・ユーザ・アプリケーション間の通信を可能にするように働く。このようにして、各通信経路ごとに伝送リンクを専用にする場合よりも、ずっと多くのトラフィックが搬送できるように、パケット・ネットワークの通信機構の利用度が最適化される。

【0055】クラスタ化技法：

1. 修正ベルマン・フォード・アルゴリズム
ネットワークにおける最適経路を計算することが可能なアルゴリズムはいくつかある。修正ベルマン・フォード・アルゴリズム (Modified Bellmann-Ford Algorithm) は、高速パケット交換ネットワークで最も一般に使用されているものの1つである。欧州特許出願第93480030.1号、"Methods for Apparatus for Optimum Path Selection in Packet Transmission Networks"に記載されているように、このアルゴリズムは、各接続要求について下記のパラメータを用いて呼び出される。

- ・ソース・ノード・アドレス
- ・宛先ノード・アドレス
- ・その接続に必要なサービスの品質。たとえば、
- ・最大呼出しセットアップ遅延時間（実時間で処理される接続の場合に非常に重要なパラメータ）。
- ・最大端末間通過遅延時間。
- ・最大情報損失、エラー確率等。

【0056】このアルゴリズムは、ソース・ノード（またはアクセス・ノード）と宛先ノードの間の最小ホップおよび経路長を決定するために、ネットワーク・トポロジのローカル・コピーを使用する。修正ベルマン・フォード・アルゴリズムでは、ネットワークの地理的構成に関する仮定を行わない。ネットワークが完全にメッシュ化または階層化されても、同じ複雑さを必要とする。本発明の目的は、所与の接続に関して、最適経路を計算するのに必要な適格ノードの数を削減することにより、ネットワーク・トポロジを簡略化することである。

【0057】2. 経路選択の複雑性

所与のノードから他のすべてのノードへ到達するために必要な合計ホップ・カウントに、ホップごとの繰返し数を掛けた積が、最小ホップ基準による経路探索の複雑さCの指標である。Cは、ノードあたりの平均リンク数1、ネットワーク内のノードの合計数N、および所与のノードから任意の宛先に到達するための最適ホップの平均数"hopt"に比例する。

【数1】

$$C = 1 \times 1 \cdot 1 \left((N-1)/N \right)^0 + \text{(第1ホップ)} \\ 2 \cdot 1 \left((N-1)/N \right)^1 + \text{(第2ホップ)} \\ 3 \cdot 1 \left((N-1)/N \right)^2 + \text{(第3ホップ)} \\ \dots \dots \dots \\ (i+1) \cdot 1 \left((N-1)/N \right)^i + \text{(第} i+1 \text{ホップ)}$$

$C = \text{hopt} \cdot N \cdot l$ *N = ネットワーク内の総ノード数
 $= 2 \cdot \text{hopt} \cdot L$ ただし: $l = 2L/N$ L = ネットワーク内の総リンク数 0、1、
 $\text{hopt} = \text{最適経路のための平均ホップ数}$ 2、... i = ホップ数
 $l = 1 \text{ ノードあたりの平均リンク数}$ *10 ただし、

$$\text{hopt} = 1/N \times 1 \cdot 1 \left((N-1)/N \right)^0 + \text{(第1ホップ)} \\ 2 \cdot 1 \left((N-1)/N \right)^1 + \text{(第2ホップ)} \\ 3 \cdot 1 \left((N-1)/N \right)^2 + \text{(第3ホップ)} \\ \dots \dots \dots \\ (i+1) \cdot 1 \left((N-1)/N \right)^i + \text{(第} i+1 \text{ホップ)}$$

$$\text{hopt} = N/1 - \left((N-1)/N \right)^{N-1} \cdot (N/1 + N - 1)$$

Nが大きければ

$$\text{hopt} = N/1$$

(例: $l = 2$ なら $\text{hopt} = N/2$ 、 $l = N$ なら $\text{hopt} = 1$)

【0058】複雑さはネットワーク内のリンクの総数に直接関係する。2ノード間の所与の経路探索について、経路の計算に使用できるリンク数を制限することにより、複雑さを極めて大きな割合で減少させることができる。この方法は、最小ホップ・カウントの制約を同じ割合で減少させない状態のときに効率的である。

【0059】3. ノード属性

本発明の基本的な概念は、所与のノードにおいて別のノードへの経路を構築するために考慮しなければならない物理リンクを事前選択し、経路選択時にこの事前選択されたリンクを使用することである。また、本発明はリンク・テーブルを作成し維持する方法についても詳しく記述する。

【0060】クラスタ化アルゴリズムの目的は、ノードの階層が得られるような形でネットワークのトポロジを分解することである。ノードは、「バックボーン」と呼ばれる第1の組と、「サブエリア」と呼ばれる多数の組に分類される。バックボーン・ノードとサブエリア・ノードは両方とも、いかなる制約もなしにポートおよびトランクを支援することができる。ノード属性はトポロジ・データベースに記録され、ネットワーク内の各変化(ノード/リンクの追加または抑制)ごとに更新される。

【0061】ノードの分類および事前選択は、経路選択機能を損なわないような形で行われなければならない。したがって、2ノード間の最適経路が常に計算できることを保証することが不可欠である。最適経路選択の基準は、次のようなものである。

【0062】・最小ホップ数。

・最小ホップ数を有する経路のうちで、負荷が最も少な

いもの。

【0063】その際に、以下のように仮定する。

- ・ネットワークがN個のノードとL個のリンクを含む。
- ・すべてのアクセス・ノードが、ネットワーク・トポロジに関する同じ知識を有する。

【0064】このような要件を満たすために、クラスタ化技法は次の規則に従って構築されなければならない。

【0065】(1) ソース・ノードと宛先ノードが同じサブエリアにある場合、このサブエリアに属するノードおよびこのサブエリアに直接接続されたバックボーン・ノードの間でだけ経路が計算される。したがって、最適経路はこの1組のノードに含まれるはずである。

【0066】(2) ソース・ノードと宛先ノードがバックボーンの要素である場合は、経路はバックボーンに属するノード間だけで計算される。最適経路は、バックボーンに完全に含まれるはずである。

【0067】(3) ソース・ノードと宛先ノードが2つの異なるサブエリアにある場合は、経路は次のものに属するノード間だけで計算される。

- ・「ソース・サブエリア」と呼ばれる、ソース・ノードのサブエリア。
- ・バックボーン。
- ・「宛先サブエリア」と呼ばれる、宛先ノードのサブエリア。

【0068】したがって最適経路は、ソース・サブエリア、バックボーンおよび宛先サブエリアに含まれることになる。

【0069】提案された分解アルゴリズムは、上記の条件が満たされる分解をもたらさなければならない。

【0070】4. クラスタ化の方法

クラスタ化アルゴリズムは、大きく6段階に分けられる。次にそれぞれの段階について説明する。

【0071】・段階1: 接続性によるノード分類

所与のノードに関連する接続性は、そのノードに接続さ

れたリンク数とする。入力ネットワークの各ノードごとに接続性を計算する。そして、ネットワーク・ノードおよびそれらの接続性のリストを確立する。このリストは、接続性の降順に分類される。

【0072】・段階2：接続性ツリーの構築

この段階の目的は接続性ツリーを構築することである。接続性ツリーは入力ネットワークのすべてのノードを含む非循環なグラフである。各ノードは接続性ツリーに1回だけ現れる。ツリー構築の規則は以下の通りである。

【0073】・最初に、接続性が最高のノードをツリーのルートとして選ぶ。いくつかのノードが同じ高さの接続性を有する場合は、それらのうちの1つを任意に選ぶ。

【0074】・ネットワークのノードがすべて選択するまで、次の繰返しを行う。

・現接続性ツリーの各リーフ（終端）について、接続性の降順にサブツリーを構築する。このサブツリーは、ルートとして当該リーフを有し、リーフとしてそのサブツリーのルート（当該リーフ）が接続されるすべてのネットワーク・ノードを有する。もちろん、まだ選ばれていないノードだけが考慮される。

・次いで、新しい接続性ツリーのすべてのリーフ（元のツリーと新しい全サブツリー）にアルゴリズムを反復適用する。

【0075】・段階3：ノードの分類

第3段階の目的は、ノードをバックボーンとサブエリアに分類することである。分類規則は次の通りである。

【0076】・接続性ツリーのリーフではないすべてのノードが、バックボーンの要素になる。

【0077】・接続性ツリーの「1枚だけの（alone）リーフ」である、すなわち、そのようなリーフの「親」が「子」としてそのリーフだけを有するようなすべてのノード（分離した終端を構成するノード）を、バックボーンに含める。

【0078】・バックボーンではないノードはすべてサブエリアの要素になる。同じ「親」を持つノードは同じサブエリアに属する。

【0079】・段階4：サブエリア間のリンクの検出

次に、アルゴリズムは、サブエリア間に存在するリンクを探す。すべてのリンクが走査される。2つのサブエリアが元のネットワーク内のリンクによって接続される場合は、接続性が最高の（そのリンクが発する）ノードがそのサブエリアから抜き出され、バックボーンの一部になる。2つのノードが同じ接続性を有する場合は、任意にそれらのうちの1つをサブエリアから引き抜き、バックボーンに含める。

【0080】・段階5：バックボーンからサブエリアへのリンクの分析

最後に、アルゴリズムは、バックボーン・ノードとサブエリア・ノードの間のすべてのリンクを走査する。サブ

エリア・ノードに直接接続された各バックボーン・ノードについて、アルゴリズムは以下のことを検査する。

【0081】・サブエリア・ノードの（接続性ツリーにおける）「親」が、バックボーン・ツリーに接続されているかどうか。

【0082】・サブエリア・ノードの「親」が、バックボーン・ノードの「親」に接続されているかどうか。

【0083】どちらの検査も否定の場合、そのサブエリア・ノードがマークされる。この手続きは、そのバックボーン・ノードに接続されたすべてのサブエリア・ノードに適用される。最後に、マークされたサブエリア・ノードのうちで、バックボーン・ノードとの接続容量が最高のノードを、サブエリアから取り除き、バックボーンに含める。

【0084】・段階6：特定サブエリアの除去

段階4および段階5の後、いくつかのサブエリアが空になり、またはわずかなノードしか含まないようになることがある。したがって、それらを除去することは価値がある。実際に、たとえばネットワーク管理の目的で、単にネットワーク内に非常に小さいサブエリアができることを避けたい場合がある。2個未満（ただしこの値はネットワーク設計者によって任意に決められた値）のノードを含むすべてのサブエリアを除去し、そのようなサブエリアに属するノードをバックボーンに移す。空のサブエリアは抑制されるだけである。

【0085】5. 例

図5に示したネットワークによって、このアルゴリズムは以下の段階を含むことを特徴とする。

【0086】・段階1：接続性によるノードの分類

最初の段階は、ノードおよびそれらの接続性のリストを確立することである。次いで、このリストを接続性の降順で分類する。以下のテーブルに結果を示す。

【0087】

ノード	接続性
a	7
b	6
d	5
e	5
c	2
f	2
g	2
h	2
i	2
j	2
o	2
k	1

21

l	1
m	1
n	1
p	1
q	1
r	1

【0088】たとえば、ノードaは他の7つのノードに接続されており、最高の接続性を有する。

【0089】・段階2：接続性ツリーの構築

このテーブルと図5に示したネットワーク・トポロジを使用して、接続性ツリーを構築する。そのツリーを図6に示す。ツリーのルートは、最高の接続性を有するノードaである。

【0090】・段階3：ノードの分類

アルゴリズムの提示の段階3に述べたように、ノードはバックボーンおよびサブエリアに分類される。この例では、3つのサブエリアが見つかっている。この最初の分解の原理を、図7に示す。たとえばノードj、k、l、m、nは、接続性ツリーのリーフであり、それらは同じ「親」を持つ。つまり同じサブエリアに属する。ノードdは、「一枚だけのリーフ」として定義されるものであり、このノードはバックボーンに含まれる。

【0091】・段階4：サブエリア間のリンクの検出

図8を見ると、jからoへのリンクが2つのサブエリア間のリンクであることが容易に理解できる。したがって、段階4で述べたようにノードのうちの1つのノードがバックボーンの一部になるはずである。この場合は、2つのノードが同じ接続性を有するので、任意のoがサブエリア2から除去される(図9)。

【0092】・段階5：バックボーンからサブエリアへのリンクの分析

ここで、アルゴリズムは、段階5で述べたようにサブエリアとバックボーンの間リンクを探す。図8に示すように、dはバックボーンの一部であり、fに接続される。

【0093】・dは、fの「親」であるeには接続されない。

【0094】・dの「親」であるcは、fの「親」であるeには接続されず、fがマークされる。同様に、g、h、iもマークされる。これらの考察から、dからeへの最短経路は、ノードf、g、hまたはiのうちの1つを通らなければならないことが明らかである。仮定により、この例では、伝送容量はネットワークのすべてのリンクで同じであり、サブエリア3から任意のfが除去される。これにより、経路選択機能の健全性が保証される。実際に、dとeの間でリンクを確立しなければならないと仮定すると、最短のバックボーン経路は4ホップ(d、c、b、a、e)である。fをバックボーンに含めると、最短経路は2ホップになる。これが最適経路である(図9)。

22

【0095】・段階6：特定サブエリアの除去

提案された例の特定の場合には除去すべきものがない。アルゴリズムの最終的な結果を、図9に示す。

【0096】トポロジ・データベース：トポロジ・データベースはあらゆるノードで複製され、ネットワーク・ノードおよびリンクに関する情報を含んでいる。トポロジ・アルゴリズムは、すべてのノードにおいて、ネットワークの一貫したビューを維持するために使用される。トポロジ・データベースは次の2種類の情報を含む。

・ノードおよびリンクの物理的特性などの静的情報を含むネットワークの物理的トポロジ。

・リンクの使用度。

【0097】最初のクラスのパラメータだけが本発明に関係する。

【0098】1. リンク属性

ノード属性に基づき、以下の規則に従って、各リンクは新しい属性によって修飾される。

【0099】

ノードi	ノードj	リンクij型
バックボーン	バックボーン	バックボーン
バックボーン	サブエリアX	サブエリアX
サブエリアY	バックボーン	サブエリアY
サブエリアY	サブエリアY	サブエリアY

【0100】これらは、ノードiおよびノードjの分類に応じて、それぞれのノードにより動的に構築される。

【0101】2. 定義

所与のノード(ノードi)において、任意の他のノード(ノードj)に到達するためにどのリンクを使用すべきかを決定できる単純な構造を構築しなければならない。この行列構造は、次のように定義することができる。

【0102】・ネットワークの各リンクを表す行と各ノードを表す列がある。そのように定義された構造は、「経路選択行列」と呼ばれるL×N行列である。

【0103】・行列要素Elnは次のようなブール値である。

・Eln=1(真)は、ノードiからノードnに到達するためにリンクlが使用できることを意味する。

・Eln=0(偽)は、ノードiからノードnに到達するためにリンクlが使用できないことを意味する。

【0104】3. 行列の初期化

経路選択行列は、各ノードにおいて構成時に、トポロジ・データベース内に含まれる情報から次の原理に従って自動的に計算される。

(a) リンク(または行列の行)

【0105】・ノードiのサブエリア内のリンクがすべて、(ノードi自体を除く)他のノードに到達するために使用できる。

【0106】・バックボーン・ノード間のリンクはすべて、ノードiと同じサブエリア内に位置する宛先ノードを除く、ネットワーク内の任意のノードに到達するため

に使用できる。

【0107】・他のすべてのサブエリア・リンク（ノードiのサブエリア内にないリンク）は、それらが属するサブエリアに到達するためにだけ使用できる。

【0108】（b）ノード（または行列の列）

・ノードj列において、ノードjがノードiのサブエリア内に位置しない場合は、バックボーン・リンクがすべて使用できる。

【0109】・ノードj列において、ノードjがノードiのサブエリア内に位置する場合は、バックボーン・リンクは使用できない。

【0110】・ノードj列において、ノードiのサブエリア内またはノードjのサブエリア内あるいはその両方にあるリンクがすべて使用できる。

【0111】・ノードj列において、他のリンクはすべて使用できない。

【0112】4. 行列の更新

新しいノードまたはリンクが追加されるたびに、ネットワークのあらゆるノードにあるトポロジ・データベースが更新される。これは、新しいネットワーク構成を含む制御メッセージによって行われる。クラスタ化アルゴリ*

ノード	a	b	c	d	e	f	o	g	h	i	j	k	l	m	n	p	q	r
リンク																		
Lab	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Lbc	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Lcd	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Ldf	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Lfe	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Lae	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Lbo	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Laj	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
Lak	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
Lal	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
Lam	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
Lan	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
Loj	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
Lbp	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
Lbq	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
Lbr	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
Lge	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Lhe	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Lie	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Lgd	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Lhd	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Lid	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

【0115】表記を簡略化するために、ノードiにおける行列をi項目のテーブル $T_i(k)$ ($k=1, \dots, L$)として示す。

・任意の経路にリンクkが使用可能なとき、 $T_i(k) = X$ 。

*ズムおよび関連する経路選択行列は、ネットワーク内の各変化ごとに自動的に計算し直される。

【0113】5. 例

図9に示すように、ネットワークは、構成時に、バックボーンおよび複数のサブエリアに分解される。

バックボーンのノード：a、b、c、d、e、f、o

サブエリア1のノード：j、k、l、m、n

サブエリア2のノード：p、q、r

サブエリア3のノード：g、h、i

バックボーンのリンク：Lab、Lbc、Lcd、Ldf、Lfe、Lae、Lbo（注：「Lab」でノードa、b間のリンクを表す）

サブエリア1のリンク：Laj、Lak、Lal、Lam、Lan、Loj

サブエリア2のリンク：Lbp、Lbq、Lbr

サブエリア3のリンク：Lge、Lhe、Lie、Lgd、Lhd、Lid

【0114】ノードrに関する対応する経路選択行列は次のとおりである。

【表1】

・ノードiからサブエリアS内に位置するノードへの経路を除く任意の経路にリンクkが使用可能なとき、 $T_i(k) = -S$ 。

・ノードiからサブエリアS内に位置するノードへの経路にのみリンクkが使用可能なとき、 $T_i(k) = S$ 。

【0116】この例では、ノードaからrにおけるテ
プルTiは次のような値を有する。

リンク	Ta	Tb	Tc	Td	Te	Tf	To	Tg	Th	Ti	Tj	Tk	Tl	Tm	Tn	Tp	Tq	Tr	
Lab	X	X	X	X	X	X	X	-3	-3	-3	-1	-1	-1	-1	-1	-2	-2	-2	
Lbc	X	X	X	X	X	X	X	-3	-3	-3	-1	-1	-1	-1	-1	-2	-2	-2	
Lcd	X	X	X	X	X	X	X	-3	-3	-3	-1	-1	-1	-1	-1	-2	-2	-2	
Ldf	X	X	X	X	X	X	X	-3	-3	-3	-1	-1	-1	-1	-1	-2	-2	-2	
Lfe	X	X	X	X	X	X	X	-3	-3	-3	-1	-1	-1	-1	-1	-2	-2	-2	
Lae	X	X	X	X	X	X	X	-3	-3	-3	-1	-1	-1	-1	-1	-2	-2	-2	
Lbo	X	X	X	X	X	X	X	-3	-3	-3	-1	-1	-1	-1	-1	-2	-2	-2	
Laj	1	1	1	1	1	1	1	1	1	1	X	X	X	X	X	1	1	1	
Lak	1	1	1	1	1	1	1	1	1	1	X	X	X	X	X	1	1	1	
Lal	1	1	1	1	1	1	1	1	1	1	X	X	X	X	X	1	1	1	
Lam	1	1	1	1	1	1	1	1	1	1	X	X	X	X	X	1	1	1	
Lan	1	1	1	1	1	1	1	1	1	1	X	X	X	X	X	1	1	1	
Loj	1	1	1	1	1	1	1	1	1	1	X	X	X	X	X	1	1	1	
Lbp	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	X	X	X
Lbq	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	X	X	X
Lbr	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	X	X	X
Lge	3	3	3	3	3	3	3	X	X	X	3	3	3	3	3	3	3	3	3
Lhe	3	3	3	3	3	3	3	X	X	X	3	3	3	3	3	3	3	3	3
Lie	3	3	3	3	3	3	3	X	X	X	3	3	3	3	3	3	3	3	3
Lgd	3	3	3	3	3	3	3	X	X	X	3	3	3	3	3	3	3	3	3
Lhd	3	3	3	3	3	3	3	X	X	X	3	3	3	3	3	3	3	3	3
Lid	3	3	3	3	3	3	3	X	X	X	3	3	3	3	3	3	3	3	3

【0117】ネットワーク内の2ノード間の経路は、ホ
ップ数が最小になる場合に最適と見なされる。すべての
最適経路は、ソース・ノードをルートとするツリーを作
成する。図11に、ノードrから確立されるすべての最
適経路を示す。第1レベルにはソース・ノードに隣接す
るすべてのノードが配置され、次に第2レベルには第1
レベルのノードに隣接するまだ配置されていないすべて
のノードが配置され、以下同様にすべてが終わるまで続
く。図12に、本発明によるノードrからノードgへの
経路探索を示す。前述のように、サブエリア3内に位置
するノードgに到達するための適格リンクは、テーブル
Trで定義される。

【0118】

適格	リンク	Tr	非適格	リンク	Tr
	Lab	-2		Laj	1
	Lbc	-2		Lak	1

Lcd	-2	Lal	1
Ldf	-2	Lam	1
Lfe	-2	Lan	1
Lae	-2	Loj	1
Lbo	-2		
Lbp	X		
Lbq	X		
Lbr	X		
Lge	3		
Lhe	3		
Lie	3		
Lgd	3		
Lhd	3		
Lid	3		

【0119】バックボーン・リンク、およびソース・ノ
ードと宛先ノードをバックボーンに接続するサブエリア
・リンクだけが、探索プロセスで考慮される。他のリン
クは、経路指定アルゴリズムの計算（たとえば、修正ベ
ルマン・フォード・アルゴリズム）の複雑さを削減する
経路選択に関与しない。この例では、2つの経路rba
egおよびrbcdgが、4ホップで最小ホップ制約を
満たす。

【0120】図5に示したネットワークの複雑さは、次
のように近似することができる。

27

【0121】

 $c = 2 \text{ hopt} \cdot L = 2 \times 2 \cdot 5 \times 22 = 110$

ただし、

 $\text{hopt} = 2 \cdot 5 \text{ ホップ}$
 $L = 22 \text{ リンク}$
 $N = 18 \text{ ノード}$
 $l = 2 \cdot 44 (44/18) \text{ リンク/ノード}$

【0122】平均ホップ数は、ノード間のすべての最適（最小）経路から決定される。

・1ホップ：rb、ab、aj、ak、al、am、an、ae、bo、bp、bq、bc、oj、cd、df、dg、dh、di、ei、eh、eg、ef。

・2ホップ：ra、rc、rq、rp、ro、qa、qc、qp、qo、pa、pc、po、oa、oc、jb、je、jk、jl、jm、jn、kb、ke、k

$$\text{hopt} = (1 \times 22 + 2 \times 52 + 3 \times 58 + 4 \times 21) / (22 + 52 + 58 + 21)$$

$$= 384 / 153$$

$$= 2.50 \text{ ホップ/最適経路}$$

【0123】本発明による使用可能リンクの事前選択により、最小ホップ制約は低下しない。たとえば、ノードrから見たノードgに到達するための経路指定アルゴリズムの複雑さC'を、30%近く減少させることができる。

 $C' = 2 \text{ hopt} L' = 2 \times 2 \cdot 38 \times 16 = 76$

ただし、

 $\text{hopt} = 2 \cdot 38 (186/78) \text{ ホップ}$
 $L' = 16 \text{ リンク}$

バックボーンのリンク：Lab、Lbc、Lcd、Ldf、Lfe、Lae、Lbo

サブエリア2のリンク：Lbp、Lbq、Lbr

サブエリア3のリンク：Lge、Lhe、Lie、Lgd、Lhd、Lid

 $N' = 13 \text{ ノード}$

バックボーンのノード：a、b、c、d、e、f、o

サブエリア2のノード：p、q、r

サブエリア3のノード：g、h、i

【0124】バックボーンからバックボーンへ、たとえばノードaからノードdへの接続に関しては、複雑さC''はこのとき5分の1に減少する。

 $C'' = 2 \text{ hopt} \cdot L'' = 2 \times 1 \cdot 9 \times 7 = 26$

ただし、

 $\text{hopt} = 1 \cdot 9 (40/21) \text{ ホップ}$
 $L'' = 7 \text{ リンク}$

バックボーンのリンク：Lab、Lbc、Lcd、Ldf、Lfe、Lae

Lbo

 $N'' = 7 \text{ ノード}$

バックボーン・ノード：a、b、c、d、e、f、o

【0125】最適経路探索の方法：本発明による方法

28

*l、km、kn、lb、le、lm、ln、mb、me、mn、nb、ne、ac、ai、ah、ag、af、bd、be、ci、ch、cg、cf、fg、fh、fi、gh、gi、hi、de。

・3ホップ：rd、re、rj、rk、rl、rm、rn、qd、qe、qj、qk、ql、qm、qn、pd、pe、pj、pk、pl、pm、pn、od、oe、ok、ol、om、on、jc、ji、jh、jg、jf、kc、ki、kh、kg、kf、lc、li、lh、lg、lf、mc、mi、mh、mg、mf、nc、ni、nh、ng、nf、bf、bg、bh、bi、ad、ce。

・4ホップ：rf、rg、rh、ri、qf、qg、qh、qi、pf、pg、ph、pi、of、og、oh、oi、jd、kd、ld、md、nd。

は、以下のように要約することができる。

【0126】・段階1：ノードが、ネットワーク構成をそのトポロジ・データベース内に記憶する。このデータベースは、ネットワーク内のすべてのノード間で交換される制御メッセージによって初期化され維持される。

【0127】・段階2：トポロジ・データベース内に記憶された情報から、ノードがバックボーン・ノードとサブエリア・ノードを決定する（クラスタ化アルゴリズム）。

【0128】・段階3：事前に計算したノード属性から、ノードがリンク属性（バックボーン・リンクとサブエリア・リンク）を決定する

【0129】・段階4：L×N経路選択行列を構築することにより、ノードがネットワーク内の各宛先ノード用の使用可能リンクを選択する。

【0130】・段階5：各接続要求ごとに、1組の事前選択された使用可能リンクについて経路指定アルゴリズムが計算される。

【0131】・段階6：接続時間中、事前に計算した経路に沿って、データ・パケットが宛先ノードに送られる。

【0132】実際のネットワークでの実験：本節では、修正ベルマン・フォード経路計算アルゴリズムに適用したクラスタ化技法の効率について例示する。シミュレーションは「一般的」な高速ネットワーク上で行った。基本的ネットワーク・トポロジから、新しい4つのトポロジを構築した。以下にその特徴を示す。

・基本ネットワーク（ネットワーク1）

・27ノード、50リンク

・サブエリアなし

・ネットワーク2

- ・67ノード、130リンク
- ・2サブエリア
- ・ネットワーク3
- ・107ノード、210リンク
- ・ネットワーク2からの2つのサブエリアと新しい2つのサブエリア
- ・ネットワーク4
- ・147ノード、290リンク
- ・ネットワーク3からの4つのサブエリアと新しい2つのサブエリア
- ・ネットワーク5
- ・227ノード、450リンク
- ・ネットワーク3からの6つのサブエリアと新しい4つのサブエリア

【0133】平均経路計算時間は、次の2つのアルゴリズムを用いて推定した。

- ・本発明の目的であるクラスタ化技法を用いない修正ベルマン・フォード・アルゴリズム。
- ・上述のクラスタ化技法を用いた修正ベルマン・フォード・アルゴリズム。

その結果を図10に示す。ノード数が増えると、クラスタ化技法を用いないアルゴリズムの性能は大幅に低下する。一方、クラスタ化技法を使用するアルゴリズムははるかに良好な性能をもたらす。性能は1秒間あたりの接続要求数（または1秒間あたりの呼出し数）で表され、複雑さCに反比例する。

【0134】

【発明の効果】本発明により、経路指定アルゴリズムの最適化基準を下げずに、またネットワーク上に追加の制御メッセージを生成せずに経路選択を高速化することが可能となる。

【0135】まとめとして、本発明の構成に関して以下の事項を開示する。

(1) 伝送リンクと相互接続された複数のノードを含むパケット交換通信ネットワークにおいて、該ネットワークのネットワーク・アクセス・ノードが、データ・パケットを受信し送信する手段と、ネットワーク構成を記憶する手段とを含むとともに、ネットワーク内に位置する各宛先ノード用の1組の使用可能リンクを自動的に事前選択する手段と、前記使用可能リンクの事前選択情報を局所的に記憶する手段と、前記アクセス・ノードと宛先ノードの間の接続要求ごとに、前記事前選択された使用可能リンクから指定経路を決定する手段とを含む、通信ネットワーク。

(2) 前記リンク事前選択手段が、前記ネットワークを1組のバックボーン・ノードと複数のサブエリア・ノードとに分解する手段を含む、上記(1)に記載のネットワーク・ノード。

(3) 前記分解手段が、前記記憶手段内の前記ネットワーク構成にアクセスする手段と、各ネットワーク・ノ

ードに接続されたリンクの数である接続性に応じて、すべてのネットワーク・ノードを分類する手段と、すべてのネットワーク・ノードを含む非循環グラフであって、各ノードが一度だけ現れ、その起点（ルート）が最高の接続性をもつノードのうちから選ばれる、接続性ツリーを構築する手段と、前記接続性ツリーの終端ノードではないノードをバックボーン・ノードにし、前記接続性ツリーの分離した終端を構成するノード（終端だけを「子」として有する該終端の「親」）を該バックボーンに含ませ、同じサブエリアに属する同じ「親」を有する、バックボーン・ノードではないノードをサブエリア・ノードにして、前記ノードをバックボーン・ノードとサブエリア・ノードに分類する手段と、2つのサブエリア間のすべてのリンクを検出し、該リンクのそれぞれを接続する2つのノードのうちで接続性が最高のノードをバックボーンに組み入れる手段と、バックボーンからサブエリアへのリンクを分析し、接続性ツリーにおける前記サブエリア・ノードの「親」がバックボーン・ノードに接続されるという条件と、前記サブエリア・ノードの「親」がバックボーン・ノードの「親」に接続されるという条件とを両方とも満たさないサブエリア・ノードを決定し、同じサブエリアにおける前記条件を満たさないすべてのノードのうちの1つを事前に定義された基準に従ってバックボーンに組み入れる手段と、空のサブエリア、もしくは事前定義されたノード数より少ないノードを含むサブエリアまたはその両方を除去する手段とを含む、上記(2)に記載のネットワーク・ノード。

(4) 前記リンク事前選択手段が、2つのバックボーン・ノードを接続するリンクであるバックボーン・リンクを決定する手段と、同じサブエリア内の2つのノードまたはサブエリア・ノードとバックボーン・ノードを接続するリンクである、サブエリア・リンクを決定する手段とを含む、上記(2)または(3)に記載のネットワーク・ノード。

(5) 前記リンク事前選択手段が、(i)宛先ノードがアクセス・ノードと同じサブエリア内に位置するときはアクセス・ノード・サブエリアに属するリンク、(ii)宛先ノードもしくはアクセス・ノードの一方がバックボーン内に位置するときは該当するノードのサブエリアに属するリンクおよびバックボーン・リンク、(iii)アクセス・ノードと宛先ノードが共にバックボーン内に位置するときはバックボーン・リンク、(iv)アクセス・ノードと宛先ノードが異なるサブエリア内に位置するときはバックボーン・リンクとアクセス・ノード・サブエリアに属するリンクと宛先ノードに属するリンクを、経路決定に使用可能なものとして選択する手段と、他のすべてのリンクを、経路決定に使用可能でないものとして選択する手段とを含む、上記(4)に記載のネットワーク・ノード。

(6) 前記指定経路決定手段が、修正ベルマン・フォー

ド・アルゴリズムを計算する手段を含む、上記(1)、(2)、(3)、(4)または(5)に記載のネットワーク・ノード。

(7) アクセス・ノード(300)において、伝送リンク(209)と相互接続された複数のノード(201ないし208)とを含むパケット交換通信ネットワーク(200)内で指定経路を選択する方法であって、ネットワーク構成を記憶する段階を含み、ネットワーク内に位置する各宛先ノードのために1組の使用可能リンクを自動的に事前選択する段階と、前記使用可能リンクの事前選択情報を局所的に記憶する段階と、前記アクセス・ノードと宛先ノードの間の接続要求ごとに、前記事前選択された使用可能リンクから指定経路を決定する段階とを含む方法。

(8) 前記事前選択段階が、前記ネットワークを1組のバックボーン・ノードと複数のサブエリア・ノードとに分解する段階を含む、上記(7)に記載の方法。

(9) 前記分解段階が、前記記憶手段内の前記ネットワーク構成にアクセスする段階と、各ネットワーク・ノードに接続されたリンクの数である接続性に応じて、すべてのネットワーク・ノードを分類する段階と、すべてのネットワーク・ノードを含む非循環グラフであって、各ノードが一度だけ現れ、その起点(ルート)が最高の接続性をもつノードのうちから選ばれる、接続性ツリーを構築する段階と、前記接続性ツリーの終端ノードではないノードをバックボーン・ノードにし、前記接続性ツリーの分離した終端を構成するノード(終端だけを「子」として有する該終端の「親」)を該バックボーンに含ませ、同じサブエリアに属する同じ「親」を有する、バックボーン・ノードではないノードをサブエリア・ノードにして、前記ノードをバックボーン・ノードとサブエリア・ノードに分類する段階と、2つのサブエリア間のすべてのリンクを検出し、該リンクのそれぞれを接続する2つのノードのうちで接続性が最高のノードをバックボーンに組み入れる段階と、バックボーンからサブエリアへのリンクを分析し、接続性ツリーにおける前記サブエリア・ノードの「親」がバックボーン・ノードに接続されるという条件と、前記サブエリア・ノードの「親」がバックボーン・ノードの「親」に接続されるという条件を両方とも満たさない、サブエリア・ノードを決定し、同じサブエリアにおける前記条件を満たさない、すべてのノードのうちの1つを事前に定義された基準に従ってバックボーンに組み入れる段階と、空のサブエリアもしくは事前定義されたノード数より少ないノードを含むサブエリアまたはその両方を除去する段階と、を含む、上記(8)に記載の方法。

(10) 前記リンク事前選択段階が、2つのバックボーン・ノードを接続するリンクであるバックボーン・リンクを決定する段階と、同じサブエリア内の2つのノードまたはサブエリア・ノードとバックボーン・ノードを接

続するリンクであるサブエリア・リンクを決定する段階とを含む、上記(8)または(9)に記載の方法。

(11) 前記リンク事前選択段階が、(i)宛先ノードがアクセス・ノードと同じサブエリア内に位置するときはアクセス・ノード・サブエリアに属するリンク、(ii)宛先ノードもしくはアクセス・ノードの一方がバックボーン内に位置するときは該当するノードのサブエリアに属するリンクおよびバックボーン・リンク、(iii)アクセス・ノードと宛先ノードが共にバックボーン内に位置するときはバックボーン・リンク、(iv)アクセス・ノードと宛先ノードが異なるサブエリア内に位置するときはバックボーン・リンクとアクセス・ノード・サブエリアに属するリンクと宛先ノードに属するリンクを、経路決定に使用可能なものとして選択する段階と、他のすべてのリンクを、経路決定に使用可能でないものとして選択する段階とを含む、上記(10)に記載の方法。

(12) 前記経路決定段階が、修正ベルマン・フォード・アルゴリズムを計算する段階を含む、上記(7)、(8)、(9)、(10)または(11)に記載の方法。

【0136】<付録>

経路指定の保全性：この節の目的は、本発明に従って分解されたネットワークにおいて、次のステートメントが考慮されることを実証することである。

1. 同じサブエリアに2つのノードが属する場合は、所与の基準に関して、これらのノード間の最適経路はこのサブエリアに属する。
2. 2つの異なるサブエリアの2ノード・メンバ間の最適経路は、その2つのサブエリアとバックボーンに完全に含まれる。
3. 2つのバックボーン・ノード間の最適経路は、そのバックボーンに完全に含まれる。

このアルゴリズムの証明は3つの部分で達成される。いくつか形式的表記法を導入した後で、前記の各ステートメントについて証明を行う。

【0137】表記法：ネットワーク・トポロジを示すグラフを $N(A, \Gamma)$ とする。 A はノードの集合であり、 Γ は A の各要素について、その要素が接続される先のノードを与える写像である。ネットワーク $N(A, \Gamma)$ に関連する接続性ツリーを $T(A, \Delta)$ とする。 Δ は Γ と同様に定義される。

定義1： $\forall a, b \in A$ であって、ネットワーク内の a から b への最小ホップ経路を $H^*(a, b) \subset A$ とする。 a から b へのホップの最小数は、 $h^*(a, b) = \text{card}\{H^*(a, b)\}$ である。

定義2： $\forall a, b \in A$ であって、 a から b への接続性ツリーの距離を $d(a, b)$ とする。 $b \in \Delta^0 \cup \Delta^{-n}(a)$ の場合、 $d(a, b) = n + m$ である。

定義3： $\forall a \in A$ であって、 a の親を $b = p(a) \in A$

とする。もちろん $a \in \Delta(b)$ である。

定義4: $N(A, \Gamma)$ のバックボーン・ノードを $B \subset A$ とする。

定義5: サブエリア i のノードを $S_i \subset A$, $i \in N$ とする。定義を容易にするために、 S_i のすべてのノードの親がそのサブエリアとバックボーンの一部である、バックボーン・ノードを考える。

【0138】＜サブエリア内で＞

命題1: $\forall a, b \in S_i$ であって、 $d(a, b) = 2$ (i) ならば $h^*(a, b) \leq 2$ (ii)

証明: (i) はツリーの基本特性によって与えられる。

(i) が与えられると (ii) は自明である。

命題2: $\forall a, b \in S_i$ であって、 $H^*(a, b) \subset S_i$

証明: $H^*(a, b) \subset S_i$ でないと仮定すると、 $c \in S_i$ とならないような $c \in H^*(a, b)$ が存在することになる。したがって、 $h^*(a, b) \geq 2$ である。命題1により、 $h^*(a, b) \leq 2$ である。したがって、解は2つだけ考えられる。

・ $h^*(a, b) = 2$ のとき、(命題1により) $H^*(a, b) = \{a, d, b\}$ となるような $d \in S_i$ が存在する。さらに d は、バックボーン内へのサブエリア S_i のエントリ点である。

・ $h^*(a, b) = 1$ のとき、 c は存在しない。

したがって、最初の命題は真でない。命題2は、所与のサブエリア内において、2ノード間の最適経路がそのサブエリア内に含まれることを示す。ただし、この経路はサブエリアの一部分であるバックボーン・エントリ点を含むことができる。

【0139】＜サブエリアから別のサブエリアへ＞2つの異なるサブエリアの2つのノード・メンバ間の最適経路は、この2つのサブエリアとバックボーンに含まれるはずである。したがって、 $\forall a \in S_i, \forall b \in S_j, i \neq j$ であって、次の関係式が満たされるはずである。

・ $H^*(a, b) \subset \{S_i \cup B \cup S_j\}$ (i)

・ $H^*(a, b) \cap B \neq \emptyset$ (ii)

関係式 (i) は、最適経路が i と j を除く任意のサブエリアを横切ることを暗示する。したがって、 $\forall k \neq i \neq j$ であって、 $c \in H^*(a, b)$ となるような $c \in S_k$ が存在しないことを保証しなければならない。この点を保証するために、次の2つのステートメントが真でなければならないことが容易に示される。

【0140】1. $\forall k \neq i \neq j, \forall a \in S_i, \forall b \in S_j, \forall c \in S_k$ であって、 $c \cup \Gamma(a) \neq \emptyset$ かつ $c \cup \Gamma(b) \neq \emptyset$ である。言い換えると、サブエリア k とサブエリア i および j との間に任意の直接リンクがなくはない。この点が保証されれば、

・ 2つのサブエリア間で直接に、あるいは

・ バックボーンを横切って、 a と b 間の経路が確立できる。

第2の場合、バックボーン内の最適経路はバックボーン

内に完全に含まれるはずである。これにより第2の点もたらされる。

【0141】2. $\forall d, e \in B$ かつ $d, e \in H^*(a, b)$ であって、 $H^*(d, e) \subset B$ である。この点の証明を次に行う。関係式 (i) が真ならば、関係式 (ii) は、サブエリア i とサブエリア j の間に任意の直接リンクがあることを暗示する。要約すると、

・ 関係式 (i) は、 $k \neq i \neq j$ のとき、サブエリア k とサブエリア i および j の間に任意の直接リンクがあることを暗示する。

・ 関係式 (ii) は、サブエリア i および j の間に任意の直接リンクがなければならないことを暗示する。

したがって、2つの異なるサブエリアのメンバである2ノード間の最適経路が、バックボーンとその2つのサブエリア内に含まれることを保証するには、このサブエリア間に任意の直接リンクがなければならない。これによりアルゴリズムの第4段が正当化される。

【0142】＜バックボーン内＞バックボーンの2ノード・メンバ間の接続を確立しなければならないと仮定する。最適経路がバックボーン内に含まれることを保証しなければならない。言い換えると、 $\forall a, b \in B$ かつ $\forall i, \forall c \in S_i$ であって、 $c \in H^*(a, b)$ でない。 a, b, c が与えられているとして、 $d(a, b) > d(c, b) + 1$ かつ $h^*(a, c) = 1$ であると仮定する。この場合、 b が a からバックボーンを介してわずか $d(c, b) + 1$ ホップで到達できない場合、明らかに $c \in H^*(a, b)$ である。そのような状況をどのように容易に検出するかが問題である。接続性ツリーが構築される方式を見てみよう。各段階で、新しいレベルが追加されるとき、実際のツリーのリーフに接続された(まだ選択されていない)すべてのノードがツリーに含まれる。したがって、2つのノードが接続される場合、接続性ツリーにおけるそれらの深さの絶対的な差は1より大きくない。したがって、バックボーンを介してわずか $d(c, b) + 1$ ホップで a を b に接続するための唯一の方法は次の場合だけとなる。

・ a が c の「親」に接続される場合。

・ a の「親」が c の「親」に接続される場合。

・ a の「親」と c の「親」に前述の2つのステートメントが反復適用される場合。

実施上の問題については、接続性の最大深さが一般的に3より小さいことを考慮しなければならない。さらに、サブエリアとバックボーン間のこの種の「特殊」リンクは極めてまれである。したがって、この検査を a, c およびそれらの「親」だけに制限することができる。この制限により証明が損なわれないことに留意されたい。実際に、検査の局所的制限は、非常に厳しい条件である。ただし、それは次善の分解をもたらすことがある。たとえばバックボーン・リンクが接続性ツリー内のより高いレベルにある場合、おそらく c をバックボーン内に移動す

35

る必要はない。

【0143】したがって、目的を達成するためには、
 $(\Gamma(a) \cap B) \cap (\Gamma(p(c)) \cap B) \neq 0$ でない
 ことを確認しなければならないだけである。このステ
 ートメントが偽である場合、cはバックボーン内に移動さ
 れる。これにより、アルゴリズムの第5段階が正当化さ
 れる。

【0144】結論：本節では、ネットワークのクラスタ
 化が適用されるとき、このアルゴリズムはネットワーク
 の2ノード間の最適経路の計算を保証することを示し
 た。

【0145】(1) ツリー構築の疑似コード

```
--セルの宣言
type CELL
type LINK is access CELL
type CELL is
record
  NAME      :STRING(20);--ノードの名前
  connectivity:INTEGER;--ノードの接続性
  LEAVES    :array(POSITIVE range())of LINK;--ボイ
  ンタのアレイ
end record;
--ノード・リストの宣言
type NODE is
record
  NAME      :STRING(20);
  connectivity:INTEGER;
  TAKEN     :BOOLEAN:=FALSE;--どのノードが考慮され
  たかを知る
end record;
NODELIST : array(POSITIVE range()) of NODE;
--NODELISTがすでに初期設定され分類されて
  いると仮定する
--NODELISTが大域変数である
procedure ConstructTree(ROOT:inout LINK);
TAKEN :INTEGER:=1; --ルートはをすでに獲得済み！
!
while(TAKEN<Number of nodes in the network)loop
AddLevel(ROOT<TAKEN);
end loop;
end ConstructTree;
procedure AddLevel(ROOT:inout LINK, TAKEN:inout INT
EGER);
i      :INTEGER;
NbLeaves:INTEGER :=0;
begin
--現ツリーのすべてのリーフを調べる・・・
if (ROOT.NbLeaves>0)then
begin
--これはリーフではない
```

36

```
for i in 0..NbLeaves loop
AddLevel(ROOT.LEAVES(i), TAKEN);
end loop;
end;
else
begin
--これはリーフである・・・
--サブツリーを構築する
for i in all indexes of NODELIST loop
if LinkExists (NODELIST(i).NAME, ROOT.NAME)then
if NODELIST(i).TAKEN=FALSE then
begin
--良い候補である
new(ROOT.LEAVES(NbLeaves));
ROOT.LEAVES(NbLeaves).NAME:=NODELIST(i).NAME;
ROOT.LEAVES(NbLeaves).connectivity:=NODELIST(i).co
nnectivity
NODELIST(i).TAKEN:=TRUE;
NbLeaves:=NbLeaves+1;
TAKEN:=TAKEN+1;
end;
end if;
end if;
end loop;
end;
end if;
end ConstructTree;
--この手順への最初の呼出しは次のような形をとる
new(ROOT);
30 ROOT.NAME:=NODELIST(0).NAME;
ROOT.connectivity:=NODELIST(0).connectivity;
ConstructTree(ROOT);
【0146】(2) ノード分類の疑似コード
--セルの宣言
type CELL
type LINKis access CELL
type CELL is
record
  NAME      :STRING(20);--ノードの名前
  connectivity:INTEGER;--ノードの接続性
  SUBAREA   :INTEGER;--ノードが属するサブエリア
  NbLeaves  :INTEGER;--このノードのリーフ数
  LEAVES    :array(POSITIVE range()) of LINK;
end record;
--ここで、ツリーが構築され、各セルのすべてのフィ
 ールドが、
--正しく初期設定されていると仮定する
--サブエリア0はバックボーンを意味する
procedure Classify(ROOT:inout LINK, SUBAREA:inout I
50 NTEGER)
```

37

```

GlobalLeaves:=INTEGER:=0;
i      :INTEGER;
Begin
  --このルートに何枚の大域リーフが接続されているか
  数える
  for i in 0..ROOT.NbLeaves loop
  if ROOT.LEAVES(i)/=null then
  if ROOT.LEAVES(i).LEAVES=0 then
  --これは大域ツリーのリーフである
  GlobalLeaves:=GlobalLeaves+1;
  end if;
  end if;
  end loop;
  --現サブツリーのすべてのリーフを走査しマークする
  for i in 0..ROOT.NbLeaves loop
  if ROOT.LEAVES(i)/=null then
  if ROOT.LEAVES(i).LEAVES=0 then
  if GlobalLeaves>1 then
  --これは大域ツリーのリーフであり単独ではない
  --それをサブエリアとしてマークする
  ROOT.LEAVES(i).SUBAREA:=SUBAREA;
  else
  --単独である
  ROOT.LEAVES(i).SUBAREA:=0
  end if;
  else
  --これはサブツリーのルートである
  --バックボーンとしてマークする
  ROOT.LEAVES(i).SUBAREA=0;
  end if;
  end if;
  end loop;
  --サブエリアが存在する場合、サブエリア番号を増分
  する
  if GlobalLeaves>1 then
  SUBAREA:=SUBAREA+1;
  end if;
  --反復的呼出し
  for i in all indexes corresponding to roots of sub
  -trees loop
  Classify(ROOT.LEAVES(i),SUBAREA);
  end loop;
  end Classify;
  --この手順への最初の呼出しは次のような形をとる
  Classify(ROOT,1);--もちろんこれは接続性ツリーの

```

38

ルートである

【図面の簡単な説明】

【図 1】従来の技術において L、クラインロックと F、カムーンが提唱した方法による、階層的経路指定を示す図である。

【図 2】高速ネットワークによって支援される、様々な回線スループット能力の機能において必要な処理時間（1秒あたりの命令数）を示すグラフである。

10 【図 3】本発明において特許請求されるアクセス・ノードおよび通過ノードを含む高速パケット交換ネットワークの典型的なモデルを示す図である。

【図 4】本発明による高速経路指定点を示す図である。

【図 5】ネットワークの典型的な例を示す図である。

【図 6】本発明による典型的ネットワークの接続性ツリーを示す図である。

【図 7】本発明によるノード分類の原理を示す図である。

【図 8】本発明によるノード分類の結果を示す図である。

20 【図 9】本発明によるネットワークの最終的分解を示す図である。

【図 10】本発明の実施を含むアルゴリズムと含まないアルゴリズムの性能の比較を示すグラフである。

【図 11】ノード r の最適経路ツリーを示す図である。

【図 12】本発明によるノード r からノード g に至る最適経路を示す図である。

【符号の説明】

200 高性能ネットワーク

201 ノード

30 202 アクセス・ノード

203 アクセス・ノード

204 アクセス・ノード

205 アクセス・ノード

206 ノード

207 ノード

208 ノード

209 トランク

210 アクセス

211 リンク

40 212 ユーザ・ネットワーク

213 企業サーバ

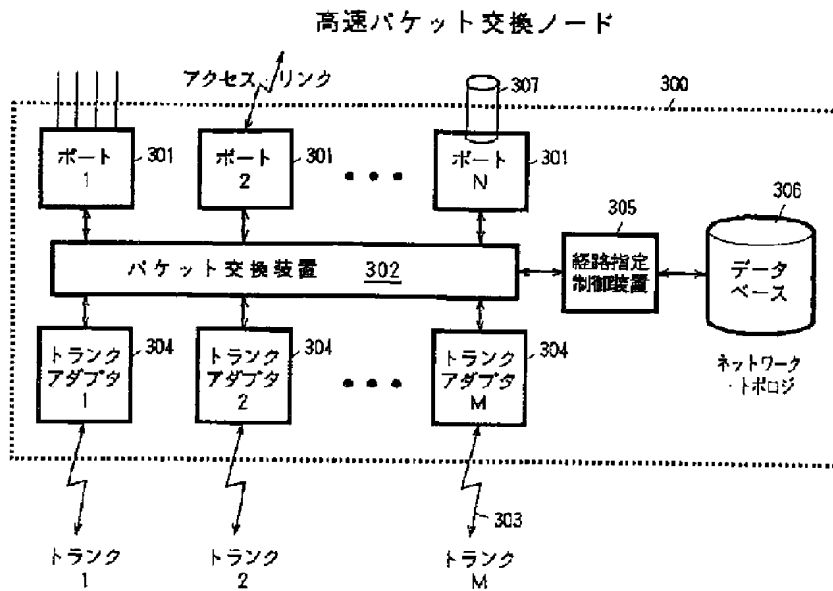
214 ローカル・エリア・ネットワーク (LAN)

215 アプリケーション・サーバ

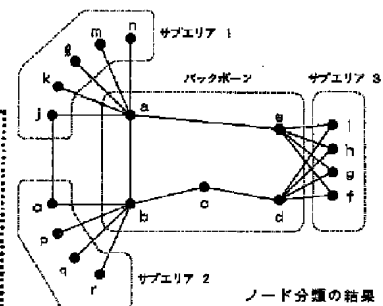
216 構内交換機 (PBX)

217 ビデオ・サーバ

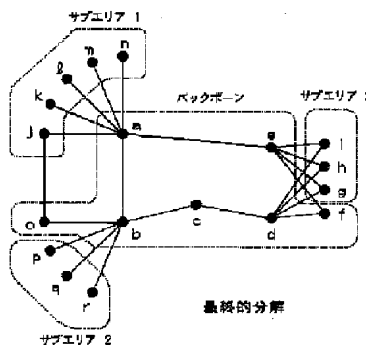
【図4】



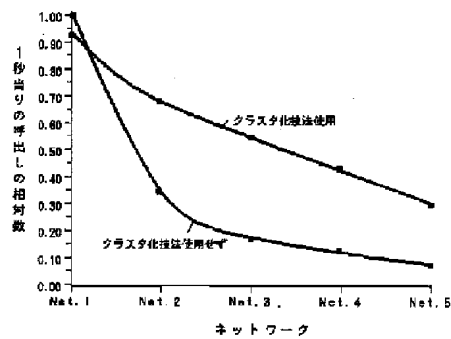
【図8】



【図9】

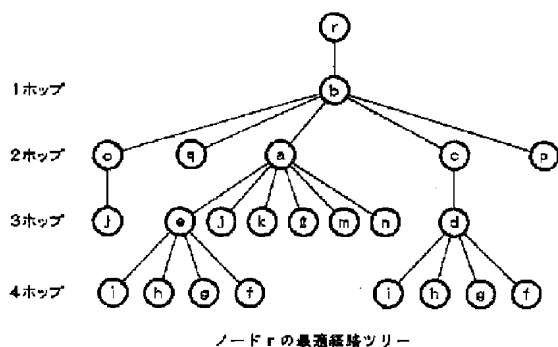


【図10】

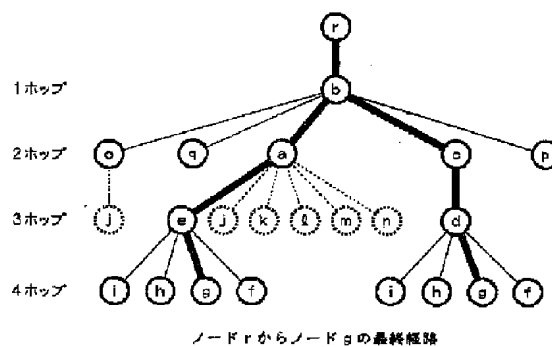


アルゴリズムの性能の比較

【図11】



【図12】



フロントページの続き

(72)発明者 バオロ・スコットン
フランス06140 ヴァンス シュマン・
ド・サント・コロンプ1561